MICROSYM
COMPUTERS INC.

131 FINCHDENE SQUARE, UNIT 5
SCARBOROUGH, ONTARIO
M1X 1A6

MICROSYM COMPUTERS INC.

EIOC (Extended I/O Controller)

Specification

Revision B, April 27 1992

PRODUCTION RELEASE - Subject to change

MICROSYM COMPUTERS INC. phone: (416)293-8263

Microsym Computers Inc. reserves the right to make changes without further notice to the products or circuits described herein to improve reliability, function, or design. Microsym Computers Inc. does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. **Microsym Computers Inc. products are not authorized for use as components in life support systems intended for surgical implant into the body or intended to support or sustain life. Buyer agrees to notify Microsym Computers Inc. of any such intended use whereupon Microsym Computers Inc. shall determine availability and suitability of its product or products for the use intended.**

Table of Contents

Specification - Extended I/O Controller

1: Description

The purpose of the Extended I/O Controller (EIOC) is to off-load and/or enhance some of the I/O functions of a host processor. This device can be used by OEM's to easily implement common I/O blocks. The EIOC is an interface co-processor chip which performs processing and data buffering for the following interfaces:

- asynchronous serial channel
- magnetic stripe card reader
- IBM® Personal Computer AT® keyboard (standard and 101 key)
- matrix keypad (up to 160 keys)
- real time clock (DS1202)
- memory mapped devices (e.g. displays)

The host processor communicates to the EIOC via a serial microprocessor interface. This interface is compatible with industry standard interfaces including ECHELON® NEUROWIRE, Motorola® SPI and National® MICROWIRE™. In addition, status signals are provided which can be monitored or used to drive interrupts.

Currently the EIOC is implemented in a 68HC705C8 OTP microcontroller, and is available in plastic 40 pin DIP and 44 pin PLCC packages. The EIOC is available in both commercial and industrial temperature ranges. In the future the EIOC will be implemented in a 68HC05C9 mask ROM microcontroller.

In addition to interface specific external components that may be needed, a clock source and some resistors and capacitors are required.

1.1: Asynchronous Serial Interface

The asynchronous serial interface is intended to provide connection to terminals, MODEMs, bar code scanners, displays, and many other devices. Full duplex communications is supported. Baud rates are software selectable from 300 to 19,200. All standard MODEM handshake signals are supported. Data formats include eight bit no parity, seven bit even parity, and seven bit odd parity. The EIOC supports 1 start bit and 1 stop bit (the EIOC can receive characters with 2 stop bits as well). There is a 96 byte receive buffer (FIFO) to allow reception of high speed bursts of data. A 32 byte transmit FIFO is provided to allow buffering of the transmit data as well. FIFO status signals are available to the host processor. The EIOC can automatically support RTS/CTS and XON/XOFF flow control without host processor intervention. The receive data ready signal can be asserted for any character received, or only when a specific character (CR for example) is received for block input applications. External line drivers and receivers are required.

1.2: Magnetic Stripe Card Reader Interface

The magnetic stripe card reader interface is intended for card access security and card identification applications. This interface supports ISO 7811 (formerly ISO 3554) track 2 readers. This is a very popular international standard. The EIOC provides automatic parity bit and LRC (Longitudinal Redundancy Check) digit verification for the card data. A 38 character data

buffer is provided. Note that the EIOC with an ISO 7811 track 2 reader will read the magnetic stripe on credit cards and A.T.M. cards. A status signal is provided for the host processor which asserts when a card has been successfully read. No external logic is required.

1.3: IBM PC/AT Keyboard Interface

The EIOC supports the bi-directional two wire IBM Personal Computer AT keyboard interface. Both standard and 101 key enhanced keyboard styles are supported. The key codes are converted to ASCII codes which can be obtained by the host processor. Function keys and special keys are converted to codes greater than 7FH. Refer to Appendix A for a summary of the output codes. A status signal is provided for the host processor to indicate that a character is ready. One 74HC05 is needed to interface to the keyboard.

1.4: Matrix Keypad Interface

The matrix keypad interface supports between 1 and 10 columns and between 1 and 16 rows for a maximum of 160 keys. The EIOC provides automatic matrix scanning and key de-bounce (50ms) The EIOC with the suggested interface circuitry will support mechanical switches and membrane switch keypads. A status signal is provided for the host processor to indicate that a key code is ready. Some external logic is required. The configuration of the external logic is dependent on the number of rows and columns needed for the switch matrix.

1.5: Real Time Clock Interface

This interface (RTC) supports the Dallas Semiconductor DS1202 serial time keeping chip. A clock tick signal is provided for the host processor which asserts once each second. The host processor can read and set the date and time at any time. In addition to the DS1202, a DS1210 non-volatile controller, one or two lithium batteries, a resistor, a capacitor, and a 32.768KHz crystal are required.

1.6: Pseudo Memory Mapped Interface

This interface which will be referred to as MMIO (Memory Mapped I/O) is used to access memory mapped devices. Examples devices that can be connected to the MMIO are displays, DTMF transceivers, A/D converters, etc. An eight bit bi-directional data bus is provided. Write (active low), read (active low), and chip select (active high or low) signals are provided. The data lines can each drive one pull-up resistor and three standard LSTTL inputs (I_{OL} = 1.6mA maximum at V_{OL} ≤0.4VDC).

Note that this is not a true memory mapped interface and is not an extension of the host processor bus or the EIOC bus. The MMIO is a "pseudo" memory mapped interface implemented in firmware in the EIOC. Since the MMIO is implemented in firmware, MMIO cycle timing is very relaxed and will extend occasionally due to EIOC internal interrupt activity.

The CTS input can be used as a "ready" input to prevent device access until the peripheral is ready. This input can be configured as active high or active low.

Two address modes are supported. Non-multiplexed address mode should be used for applications using no or only one address line. If more address lines are needed, the EIOC can be configured for multiplexed address mode. In this mode up to 8 address lines can be latched using an external address latch. An ALE signal is provided for those devices that have an internal latch.

1.7: Host Processor Interface

The host processor interface is implemented using a synchronous serial interface. This interface is compatible with Motorola SPI, ECHELON NEUROWIRE, and National MICROWIRE. The host processor must be configured as the master. Commands, status, and data are transferred via this serial interface between the host and the EIOC.

A global interrupt signal is provided for host processors. This signal is a logical OR of the status signals. Specific interrupt sources can be masked under software control. Additional hardware status signals are provided which can be used to trigger "when" clauses in ECHELON NEURON® CHIP applications. The signals provided are as follows:

- receive data ready (RRDY)
- serial status change (SSC)
- transmit buffer empty (TBE)
- transmit buffer full (TBF)
- card data ready (CDR)
- keyboard data ready (KDR)
- real time clock tick (TICK)

NOTES:

1) Not all of the interfaces described can be used simultaneously. The EIOC interface configuration must be selected after power up. Only one keyboard interface can be selected (either PC/AT or matrix). If the MMIO (Memory Mapped I/O) interface is selected then the following restrictions are placed on the device:

- the magnetic stripe card reader interface is not provided
- MODEM signals RTS, DTR, DSR, DCD, and RI are not provided
- TBE and CDR status signals are not provided

2) When using the IBM PC/AT keyboard interface the asynchronous serial interface should not be used to receive data at 19,200 baud, since this will result in intermittent receive character loss.

2: Signal Descriptions

MNEMONIC	DIP PIN	PLCC PIN	I/O	FUNCTION
VSS	20	22	in	0V power
VSS	37	41	in	0V input
VDD	40	44	in	5V power
VDD	3	4	in	5V input
SDO	31	34	out	host serial data output
SDI	32	35	in	host serial data input
SCK	33	36	in	host serial clock
SS	34	37	in	slave select
TXD	30	33	out	serial transmit data
RXD	29	32	in	serial receive data
OSC1	39	43	in	crystal oscillator input
OSC2	38	42	out	crystal oscillator output
RES	1	1	I/O	EIOC reset
ICLK	14	15	out	interface clock
ISEL	15	16	out	0=column decoder, 1=clock chip
IMFP	12	13	I/O	RTC data I/O, row mux A0, MMIO A0 or ALE
CRES	16	17	out	column decoder reset
CRES		18	out	column decoder reset
K0	2	2	in	KBD clk in, row 1 in, rows 1-8 input from mux
K1	13	14	in	KBD data in, row 2 in, rows 9-16 input from mux
K2	17	19	I/O	KBD clk out, row 3 in, row mux A1 out
K3	18	20	I/O	KBD data out, row 4 in, row mux A2 out
CTS/RDY	36	39	in	MODEM CTS in, MMIO ready in
CTS/RDY		40	in	MODEM CTS in, MMIO ready in
RTS/AD0	11	12	I/O	MODEM RTS out, MMIO address/data 0
DTR/AD1	10	11	I/O	MODEM DTR out, MMIO address/data 1
DSR/AD2	9	10	I/O	MODEM DSR in, MMIO address/data 2
DCD/AD3	8	9	I/O	MODEM DCD in, MMIO address/data 3
RI/AD4	7	8	I/O	MODEM RI in, MMIO address/data 4
CLS/AD5	6	7	I/O	card loaded, MMIO address/data 5
CD/AD6	5	6	I/O	card data, MMIO address/data 6
CCK/AD7	4	5	I/O	card clock, MMIO address/data 7
INT	19	21	out	host processor interrupt
RRDY	28	31	out	receive data ready
SSC	27	30	out	serial status change
TBF	26	29	out	transmit buffer full
KDR	25	28	out	keyboard data ready
TICK	24	27	out	clock tick
CDR/WR	23	26	out	card data ready, MMIO write
TBE/RD	22	25	out	transmit buffer empty, MMIO read
CS	21	24	out	MMIO chip select

2.1: VSS and VDD (power)

A 5VDC power supply must be connected to VSS (0V) and VDD (5V). EIOC operation is specified only at 5VDC ±10%. Do not attempt to operate the EIOC at any voltage outside of this range.

VDD and VSS should be de-coupled using a ceramic capacitor with good high frequency characteristics. Traces between the VDD pins, and between the VSS pins, on the EIOC should be as short as possible (avoid loops).

2.2: OSC1 and OSC2 (oscillator)

Use of a resistor To generate the clock frequency is not recommended for the EIOC. The oscillator frequency for the EIOC is specified at 3.6864MHz. Note that if the serial asynchronous interface is not being used, then this exact frequency is not needed. When not using the serial asynchronous interface, a higher frequency (maximum 4Mhz) can be used if desired. A slower frequency may be acceptable under certain conditions. For example, if the EIOC is being used for the magnetic stripe card reader interface only, then it would be acceptable to operate the EIOC at 2.5MHz. When running at frequencies lower than 3.6864 Mhz, the delay specified for read commands (READ DELAY) will have to be scaled up accordingly. Please feel free to contact Microsym for applications assistance. Acceptable clock sources for the EIOC are as follows:

1) Crystal: A 3.6864MHz parallel resonant crystal must be connected to OSC1 and OSC2. Suitable crystals include (but are not limited to) M-Tron MP-1-3.686400, Fox FOX0368-20, CTS® MP037 and NYMPH® NYP037-20. A 4.7MΩ bias resistor and two 39pf load capacitors are needed as well.

2) Ceramic Resonator: A 3.69MHz ceramic resonator can be used if the frequency and temperature tolerances combined do not exceed ±1%. The resonator manufacturer should be consulted to determine proper load capacitor and series resistor values.

3) HCMOS signal: In this case the signal output is connected to OSC1 and OSC2 is not connected. The minimum pulse width of the frequency source (T_{OH} or T_{OL}) is 90ns.

The oscillator components should be located as close to the EIOC as possible. The ground connections for the load capacitors should be made as short as possible between the capacitors and the EIOC VSS pins.

2.3: RES (RESet)

The EIOC generates its own power on reset. An external device can drive the open drain EIOC RES pin (active low) if desired. The reset duration must be at least 5µs. The EIOC internal watchdog and clock monitor will drive RES low for about 3µs if a time-out condition occurs. When a time-out occurs, the EIOC will perform a normal power on reset and must be re-configured. It is recommended that the host processor reset be tied to the EIOC reset so that a watchdog time-out condition in either processor will reset both the host processor and the EIOC.

2.4: SDO, SDI, SCK, and SS (host interface)

The host processor serial interface is implemented using these pins. The host processor must be configured as the master. SDO is a Serial Data Output which transfers data from the EIOC to the host. SDI is a Serial Data Input which transfers data to the EIOC from the host. SCK is the Serial ClocK signal which is always generated by the host. The host must drive SS (Slave Select) low during the transfer.

2.5: ICLK, ISEL (Interface CloCK, Interface SELect)

ICLK is generated by the EIOC to clock both the column driver and the real time clock. When ISEL is high, data is clocked into or out of the real time clock with ICLK. When ISEL is low and the EIOC is configured for a matrix keypad, ICLK advances the 74HC4017 column driver to the next column.

2.6: IMFP (Interface Multi-Function Pin)

The function of IMFP changes depending on the EIOC configuration and the I/O device being accessed.

When accessing the real time clock, IMFP is used to transfer data between the EIOC and the real time clock.

When the EIOC is configured for matrix keyboard with greater than 4 rows, IMFP is used to output A0 for the 74HC151 row input multiplexer(s).

When the EIOC is configured for memory mapped I/O and non-multiplexed address/data, IMFP provides A0 (address 0) for the memory mapped peripheral.

When the EIOC is configured for memory mapped I/O and multiplexed address/data, IMFP provides ALE (address latch enable) for the address latch.

2.7: CRES (Column decoder RESet)

When the EIOC is configured for a matrix keypad, CRES is used to reset the 74HC4017 column decoder to column 0.

Note that on the PLCC packages the pin connection is different for a 68HC705C8 than for a 68HC05C9. To ensure design compatibility for both devices, both pin 17 and pin 18 should be connected together. One of these pins will be the CRES output and the other will not be connected internally to the EIOC.

2.8: K0 (keyboard interface)

When the EIOC is configured for an IBM PC/AT keyboard, K0 (Keyboard interface signal 0) is the keyboard clock input.

When the EIOC is configured for a matrix keypad with 4 or less row inputs, K0 is the row 1 input. Refer to the evaluation board schematic for information on how to properly terminate this signal.

When the EIOC is configured for a matrix keypad with 5 or more row inputs, K0 connects to the 74HC151 multiplexer output for rows 1-8.

2.9: K1 (keyboard interface)

When the EIOC is configured for an IBM PC/AT keyboard, K1 (Keyboard interface signal 1) is the keyboard data input.

When the EIOC is configured for a matrix keypad with 4 or less row inputs, K1 is the row 2 input. Refer to the evaluation board schematic for information on how to properly terminate this signal.

When the EIOC is configured for a matrix keypad with between 5 and 8 row inputs, K1 is not used and should by tied high with a 10KΩ to 47KΩ pull-up resistor.

When the EIOC is configured for a matrix keypad with 9 or more row inputs, K1 connects to the 74HC151 multiplexer output for rows 9-16.

2.10: K2 (keyboard interface)

When the EIOC is configured for an IBM PC/AT keyboard, K2 (Keyboard interface signal 2) is the keyboard clock output.

When the EIOC is configured for a matrix keypad with 4 or less row inputs, K2 is the row 3 input. Refer to the evaluation board schematic for information on how to properly terminate this signal.

When the EIOC is configured for matrix keyboard and greater than 4 rows, K2 is used to output A1 for the 74HC151 row input multiplexer(s).

2.11: K3 (keyboard interface)

When the EIOC is configured for an IBM PC/AT keyboard, K3 (Keyboard interface signal 3) is the keyboard data output.

When the EIOC is configured for a matrix keypad with 4 or less row inputs, K3 is the row 4 input. Refer to the evaluation board schematic for information on how to properly terminate this signal.

When the EIOC is configured for matrix keyboard and greater than 4 rows, K3 is used to output A2 for the 74HC151 row input multiplexer(s).

2.12: CTS/RDY (Clear To Send/Ready)

When the EIOC is not configured for memory mapped I/O, this signal is the active low CTS signal for the MODEM port. The EIOC can be configured for RTS/CTS flow control which means that the EIOC signal will not transmit serial data when the CTS input is high. If RTS/CTS flow control is not used then CTS is a general purpose bit input.

When the EIOC is configured for memory mapped I/O, CTS/RDY is a ready input. In this mode the EIOC can be configured for ready being active high or active low. Neither the serial port (TXD) nor the memory mapped I/O port will output data when the RDY input is negated.

Note that on the PLCC packages the pin connection is different for a 68HC705C8 than for a 68HC05C9. To ensure design compatibility for both devices, both pin 39 and pin 40 should be connected together. One of these pins will be the CTS/RDY input and the other will not be connected internally to the EIOC.

2.13: RTS/AD0 (Request To Send/Address & Data 0)

When the EIOC is not configured for memory mapped I/O, this signal is the active low RTS signal for the MODEM port. The EIOC can be configured for RTS/CTS flow control, and in this mode the connected equipment should not begin to transmit data to the EIOC when the RTS output is high. If RTS/CTS flow control is not selected then RTS is a general purpose bit output.

When the EIOC is configured for memory mapped I/O and non-multiplexed address & data, RTS/AD0 is bi-directional data bit 0. If multiplexed address & data is selected, RTS/AD0 is multiplexed address bit 0 output and bi-directional data bit 0.

2.14: DTR/AD1 (Data Terminal Ready/Address & Data 1)

When the EIOC is not configured for memory mapped I/O, this signal is a general purpose bit output (DTR) intended for MODEM control.

When the EIOC is configured for memory mapped I/O and non-multiplexed address & data, DTR/AD1 is bi-directional data bit 1. If multiplexed address & data is selected, DTR/AD1 is multiplexed address bit 1 output and bi-directional data bit 1.

2.15: DSR/AD2 (Data Set Ready/Address & Data 2)

When the EIOC is not configured for memory mapped I/O, this signal is a general purpose bit input (DSR) intended for MODEM control.

When the EIOC is configured for memory mapped I/O and non-multiplexed address & data, DSR/AD2 is bi-directional data bit 2. If multiplexed address & data is selected, DSR/AD2 is multiplexed address bit 2 output and bi-directional data bit 2.

2.16: DCD/AD3 (Data Carrier Detect/Address & Data 3)

When the EIOC is not configured for memory mapped I/O, this signal is a general purpose bit input (DCD) intended for MODEM control.

When the EIOC is configured for memory mapped I/O and non-multiplexed address & data, DCD/AD3 is bi-directional data bit 3. If multiplexed address & data is selected, DCD/AD3 is multiplexed address bit 3 output and bi-directional

data bit 3.

2.17: RI/AD4 (Ring Indicator/Address & Data 4)

When the EIOC is not configured for memory mapped I/O, this signal is a general purpose bit input (RI) intended for MODEM control.

When the EIOC is configured for memory mapped I/O and non-multiplexed address & data, RI/AD4 is bi-directional data bit 4. If multiplexed address & data is selected, RI/AD4 is multiplexed address bit 4 output and bi-directional data bit 4.

2.18: CLS/AD5 (Card Loaded Signal/Address & Data 5)

When the EIOC is not configured for memory mapped I/O, this signal is the active low CLS (card loaded signal) input from the magnetic stripe card reader.

When the EIOC is configured for memory mapped I/O and non-multiplexed address & data, CLS/AD5 is bi-directional data bit 5. If multiplexed address & data is selected, CLS/AD5 is multiplexed address bit 5 output and bi-directional data bit 5.

2.19: CD/AD6 (Card Data/Address & Data 6)

When the EIOC is not configured for memory mapped I/O, this signal is the CD (card data) input from the magnetic stripe card reader.

When the EIOC is configured for memory mapped I/O and non-multiplexed address & data, CD/AD6 is bi-directional data bit 6. If multiplexed address & data is selected, CD/AD6 is multiplexed address bit 6 output and bi-directional data bit 6.

2.20: CCK/AD7 (Card ClocK/Address & Data 7)

When the EIOC is not configured for memory mapped I/O, this signal is the CCK (card clock) input from the magnetic stripe card reader.

When the EIOC is configured for memory mapped I/O and non-multiplexed address & data, CCK/AD7 is bi-directional data bit 7. If multiplexed address & data is selected, CCK/AD7 is multiplexed address bit 7 output and bi-directional data bit 7.

2.21: INT (INTerrupt)

This signal is an active low interrupt output. This signal can be used to interrupt host processors. Interrupt sources are selected during EIOC configuration. The INT output is negated (set high) when the source of the interrupt is serviced and no other interrupt source is pending. Therefore, the host processor should use level detection and not edge detection for its

interrupt pin.

2.22: RRDY (serial Receive ReaDY)

This signal is an active high output used to provide status to the host processor.

When the EIOC is configured for RRDY on any character, this signal is asserted (set high) when one or more characters are in the asynchronous serial interface receive buffer. In this case, the RRDY signal is negated when there are no characters remaining in the receive buffer.

When the EIOC is configured for RRDY on a termination character, RRDY will not assert until the termination character specified during configuration has been received. Examples of termination characters that might be selected are CR and EOT. This mode permits the EIOC to receive entire blocks of data (up to 96 bytes) before interrupting or signaling the host processor. In this case, the RRDY signal is negated after all characters up to and including the terminating character have been read from the buffer. The EIOC can begin receiving a new block of data into the buffer while the host processor is retrieving the previous block. If another terminating character is received before the terminating character from the previous block has been retrieved by the host processor, then RRDY will not negate when the first terminating character is retrieved. Therefore, the host processor should not use the negation of RRDY to determine when a terminating character has been retrieved. Note that if the receive buffer becomes full then RRDY will assert even if the termination character has not yet been received.

If the EIOC is configured for flow control, and for RRDY on a termination character, there is a possibility that reading a termination character from the buffer will cause RRDY to negate while receiver flow is off. In this case the EIOC will send XON or assert RTS in order to enable receiver flow again. RRDY will eventually assert again as a result of another terminator being received or the receive buffer filling up.

2.23: SSC (Serial Status Change)

This signal is an active high output used to provide status to the host processor. This signal is asserted (set high) when status changes in the asynchronous serial interface. Status includes:

- CTS input change
- DSR input change
- DCD input change
- RI input change
- receive parity error
- receive framing error
- receive buffer overrun

SSC is negated when the serial status register is read by the host processor.

2.24: TBF (serial Transmit Buffer Full)

This signal is an active high output used to provide status to the host processor. This signal is asserted (set high) when the 32 character asynchronous serial interface transmit buffer is full. The host processor should not attempt to send data for transmission when TBF is high. TBF is negated when the EIOC transmits a character from the buffer.

2.25: KDR (Keyboard Data Ready)

This signal is an active high output used to provide status to the host processor. This signal is asserted (set high) when a character is available from either the PC/AT keyboard or the matrix keypad. This signal is negated when the character is retrieved by the host processor.

2.26: TICK (clock TICK)

This signal is an active high output used to provide status to the host processor. When a real time clock is connected to the EIOC, this signal asserts (goes high) once each second. This signal is negated when the the real time clock seconds are read by the host processor. Note that this signal will only operate if a real time clock chip is connected to the EIOC and configured.

Note that when the card reader interface is being used that the TICK signal will not assert while a card is being swiped through the card reader. The EIOC does not access the real time clock chip while the data is being read from the card because of the processing overhead required to ensure correct interpretation of the bit stream from the card reader.

2.27: CDR/WR (Card Data Ready/WRite)

When the EIOC is not configured for memory mapped I/O, this signal is an active high output which asserts (goes high) when a valid data stream has been read from a magnetic stripe card. This signal is negated when the entire data stream has been retrieved by the host processor. The EIOC can buffer up to 38 characters.

When the EIOC is configured for memory mapped I/O, CDR/WR is an active low write strobe output which should be used to strobe data into external devices.

2.28: TBE/RD (serial Transmit Buffer Empty/ReaD)

When the EIOC is not configured for memory mapped I/O, this signal is an active high output which asserts (goes high) when there are no characters in the 32 byte asynchronous serial interface transmit buffer. This signal negates when the host processor outputs data to the EIOC for transmission.

When the EIOC is configured for memory mapped I/O, this signal is an active low read strobe which should be used to signal external devices to output data onto the data bus.

2.29: CS (Chip Select)

When the EIOC is not configured for memory mapped I/O, this signal is a general purpose output bit. The level of this signal is defined using the write MODEM signal levels command.

When the EIOC is configured for memory mapped I/O, this signal can be used as an active high or active low output to select an external device for reading or writing. The active level of this signal is determined during EIOC configuration.

2.30: TXD (Transmit Data)

This signal is the asynchronous serial interface transmit data output. When not transmitting this output will normally be high. During reset this signal will be low (break condition).

2.31: RXD (Receive Data)

This signal is the asynchronous serial interface receive data input. If the receiver is not used then this input must be tied high.

3: Applications Information

For electrical and mechanical specifications please refer to the Motorola MC68HC705C8 technical data book (Motorola order number MC68HC705C8/D REV 1).

3.1: Matrix Keypad Optimization

In many cases the I/O lines of the host processor can be used to scan matrix keypads. Where this is not practical, or where it is desirable to reduce the amount of software overhead in the host processor, the EIOC can be used.

This section discusses the optimum column/row combinations in order to minimize external component cost. Where possible the I/O lines of the host processor should be used to scan matrix keypads. Where this is not practical, the EIOC will support matrix keypads up to 160 keys. The amount of additional circuitry needed depends on the number of keys.

For one column no external logic is required. For two or more columns, a 74HC4017 is needed to decode column outputs. After the 74HC4017, open drain drivers are used to drive each column. For 2 to 6 columns, one 74HC4017 chip and one 74HC05 chip are required. For 7 to 10 columns one 74HC4017 chip and two 74HC05 chips are required.

For scanning up to 4 rows, no external logic is required. For scanning between 5 and 8 rows, one 74HC151 multiplexer is required. For scanning between 9 and 16 rows, two 74HC151 multiplexers are required. The following table gives the optimum configurations based on the number of keys scanned:

KEYS	COLS	ROWS	74HC4017	74HC05	74HC151	
1-4	1	1-4	0	0	0	<- non-matrix
5-8	1	5-8	0	0	1	<- non-matrix
9-16	1	9-16	0	0	2	<- non-matrix
≤12	3	4	1	1	0	
≤16	4	4	1	1	0	
≤20	5	4	1	1	0	
≤24	6	4	1	1	0	
≤30	6	5	1	1	1	
≤36	6	6	1	1	1	
≤42	6	7	1	1	1	
≤48	6	8	1	1	1	
≤56	7	8	1	2	1	
≤64	8	8	1	2	1	
≤72	9	8	1	2	1	
≤80	10	8	1	2	1	
≤84	6	14	1	1	2	
≤90	6	15	1	1	2	
≤96	6	16	1	1	2	
≤112	7	16	1	2	2	
≤128	8	16	1	2	2	
≤144	9	16	1	2	2	
≤160	10	16	1	2	2	

Note that the first three listings in the table are referred to as "non-matrix". In these cases the keypad or switch group has a common connection to ground, and the other switch connections are wired to the row multiplexer inputs. This allows the use of common bus ("single pole") keypads when the total number of keys is 16 or less.

When the EIOC is powered on, it will not recognize any keyboard input. To use the PC/AT keyboard interface, the EIOC configuration register must be written with bit 0 set. To use the matrix keyboard interface, the EIOC configuration register must be written with bit 0 clear, and the matrix keypad configuration register must be written with a valid row/column combination.

Due to processor overhead, matrix keypad scanning is suspended during the time that a card is being swiped through the magnetic stripe card reader. If the magnetic stripe card reader is not used then matrix keypad scanning is continuous.

3.2: Unused Signals

All unused input or I/O signals must be tied high using 10KΩ to 47KΩ pull-up resistors.

3.3: Example Schematics

Refer to the example schematics provided for details on how to connect the various interfaces to the EIOC in different modes.

3.4: IBM PC/AT Keyboard

When designing the power supply for your application, it is important to determine and provide the necessary power (@5VDC) for the external keyboard (if used). Standard IBM PC/AT compatible keyboards include an NMOS microcontroller and three LEDs.

Note that clone keyboards that have "auto configuration" only (automatic selection between XT and AT emulation) may not work with the EIOC. Keyboards with switch selection of XT and AT (set to AT) are recommended.

When using the IBM PC/AT keyboard interface, the asynchronous serial interface should not be used to receive data at 19,200 baud. If keys are being pressed on the PC/AT keyboard while a burst of data is being received at 19,200 baud, then intermittent receive data loss will result. This occurs because the EIOC runs out of processing time and an internal task overrun condition occurs. There is no error indication from the EIOC when this happens. When using the IBM PC/AT keyboard interface it is best to limit the asynchronous serial baud rate to 9,600.

3.5: Card Reader

Suitable suppliers for inexpensive magnetic stripe card readers that are compatible with the EIOC include, but are not limited to, Neuron Electronics

Inc. (213)793-1300 and Tokin America Inc. (708)297-0070. Note that Neuron Electronics Inc. supplies the mating P.C.B. connectors with their card readers.

The EIOC firmware is designed to read track 2 (ISO 7811) data. When ordering card readers, it is important to specify track 2. Note that the EIOC does not support writing. Depending on the application, custom cards can be manufactured and encoded, or users can use their credit cards or A.T.M. cards as identification cards in a system.

For information on the encoding of the data on track 2, refer to ISO standard 7811/2 "Identification cards - Recording technique - Part 2: Magnetic stripe". Note that if you are getting cards custom made, then the first character encoded must be the start sentinel (0x0B), the second last character must be the end sentinel (0x1F), and the last character must be the LRC character. The total number of characters including the LRC character must be less than or equal to 40. If these requirements are not met then the EIOC card read algorithm will not properly handle the data from the card.

For information on the data recorded on track 2 of credit cards and A.T.M. cards, refer to ISO standard 7813 "Identification cards - Financial transaction cards".

The account number on financial transaction cards includes a "double add double" check digit. This digit is not the LRC digit. The EIOC does not verify the check digit because some custom applications will not use the standard account number (also "identification number" or "PAN - Primary Account Number") format. If you are using the standard account number format, or if you are using credit cards or A.T.M. cards, then you may want to verify the check digit to ensure that the account number is valid. The method for verifying this digit is contained in ISO standard 7812 "Identification cards - Numbering systems and registration procedure for issuer identifiers".

The EIOC does not send the start sentinel digit (0x0B) to the host. All other control characters are sent to the host as data.

Note that the EIOC strips the parity bits from the characters before sending them to the host. The EIOC automatically checks the parity bits. Also, the EIOC does not include the LRC character when transferring data to the host. The EIOC automatically verifies that the LRC is correct so there is no need to pass this on. If the EIOC detects an error in either the parity or the LRC then the data is discarded and nothing is reported to the host processor (in this case the user must swipe the card through the reader again).

If an error occurs when reading a card, the eioc will place the digits:

F000000000000000000000000000000000000000

in the buffer and then assert CDR/WR. This is a null digit string and he host processor should read this as if it was normal data from the card reader. The host processor can tell the difference between a valid digit string and this null string by checking the first digit. If the first digit is an end sentinel (0x0F) then an error was encountered when reading the card. If the first digit is not an end sentinel then the digit string is a valid string from a card.

One word of caution if you plan to use credit cards as identification cards in a system. The expiry date and credit authorization codes are encoded on track 2 as well as the account number. When users are issued new cards by the financial institutions, the expiry date will be different on the new cards. As a result, only the account number portion of the data on track 2 should be used as the identifier for the user. The account number starts after the start sentinel, and is followed by a field separator character (0x0D). The length of the account number varies, and is dependent on the issuing financial institution.

3.6: High Reliability Applications

Note that initially the EIOC will be supplied as a pre-programmed 68HC705C8 OTP (One Time Programmable) device. It is known that a small number of these devices will eventually suffer program corruption as a result of programmed bits changing state. This failure mode can be avoided by subjecting the components to a 24 hour bake (out of circuit) at 150°C and then subsequently testing them. The bake will effectively weed out the devices that are subject to this failure mode.

The EIOC includes a power on 16 bit checksum test which will verify that the program is correct. If the test fails the EIOC will halt program execution immediately. Therefore, if the host cannot communicate with the EIOC then it can safely be assumed that the EIOC is defective.

Microsym Computers Inc. products are not authorized for use as components in life support systems intended for surgical implant into the body or intended to support or sustain life. Buyer agrees to notify Microsym Computers Inc. of any such intended use whereupon Microsym Computers Inc. shall determine availability and suitability of its product or products for the use intended.

3.7: Handling Precautions

The EIOC is fabricated using a high speed CMOS process, and as such all normal CMOS handling precautions should be observed when handling the EIOC.

3.8: Asynchronous Serial Interface

When the serial configuration register is written both the receive buffer and the transmit buffer are flushed.

When using XON/XOFF flow control the number of characters in the receive buffer will dictate when XON and XOFF are sent. When the receive buffer is almost full (90 characters) the EIOC will transmit XOFF. When the host processor has reduced the number of characters in the receive buffer to 60, the EIOC will transmit XON. XON is also transmitted after the serial configuration is written and after a flush receive buffer command is issued.

When using RTS/CTS flow control the number of characters in the receive buffer will dictate when RTS is asserted or negated. When the receive buffer is almost full (90 characters) the EIOC will negate RTS (high). When the host processor has reduced the number of characters in the receive buffer to 60,

the EIOC will assert RTS (low). RTS is also asserted after the serial configuration is written and after a flush receive buffer command is issued.

The asynchronous serial interface is designed to allow full duplex communications at a maximum baud rate of 19,200. Note that data cannot be received continuously at this rate because the data transfer rate between the EIOC and the host processor is limited.

For ECHELON NEURON CHIPS, the maximum baud rate for the host processor connection (NEUROWIRE I/O) is 20,000 bits per second. Taking into account the delay needed in the read data command, the maximum number of bytes that can be read from the EIOC is around 900 characters per second. This will practically be less because of the response time of the NEURON CHIP "when" clauses, and the network bandwidth. As a result, continuous asynchronous serial interface data reception at more than 4800 baud will definitely result in loss of received data. The EIOC will store up to 96 characters in its receive buffer. This is useful for buffering small blocks of data received at high speeds. The 4800 baud figure quoted is best case. Actual maximum throughput will have to be evaluated for each application.

For other host processors continuous data reception at 19,200 baud may result in loss of data, due to the time required to transfer data from the EIOC to the host processor.

When using the IBM PC/AT keyboard interface, the asynchronous serial interface should not be used to receive data at 19,200 baud. If keys are being pressed on the PC/AT keyboard while a burst of data is being received at 19,200 baud, then intermittent receive data loss will result. This occurs because the EIOC runs out of processing time and an internal task overrun condition occurs. There is no error indication from the EIOC when this happens. When using the IBM PC/AT keyboard interface it is best to limit the asynchronous serial baud rate to 9,600.

3.9: Real Time Clock

Where it is not practical to connect a real time clock chip to the host processor data bus or I/O lines, a real time clock chip (Dallas DS1202) can be connected to the EIOC. Also, the EIOC will significantly reduce the amount of software overhead required especially when dealing with some serial interface clock chips.

The long term accuracy of the real time clock is specified by Dallas Semiconductor to be plus or minus two minutes per month at 25°C when using the recommended crystal.

Note that the real time clock will not run until the first time the time (hours, minutes and seconds) is sent to the EIOC from the host. Until then the DS1202 is in a low power standby mode where the clock does not run. The clocks on evaluation boards will be running since they are set during testing of the boards.

3.10: Host Interface

The host interface is primarily a synchronous serial data link which is used to transfer commands, status and data. This serial interface is compatible with ECHELON NEUROWIRE, Motorola SPI, and National MICROWIRE. In addition to the serial data link there is an interrupt signal and some status signals.

The interrupt signal is useful for host processors that support external hardware interrupts. The EIOC can be configured to interrupt the host processor when various EIOC events occur.

The status signals can be useful for all host processors. For processors where an external interrupt is used, the signals can be used to quickly determine the source of the interrupt. For host processors that do not support an external interrupt, these signals can be polled by the host processor to determine when an EIOC event has occurred. For example, these signals can be connected to bit inputs on ECHELON NEURON CHIPS, and "when" clauses can be used to trigger tasks when the status signals are asserted by the EIOC.

The synchronous serial interface is used to transfer commands, status and data. Commands and data are transferred most significant bit first. The commands the host uses to control the EIOC and transfer data are detailed in this section. There are three types of data transfers defined as follows:

3.10.1: Synchronization

In this case the synchronization command is written without any subsequent data transfer. See section 3.11.1: "Synchronization".

3.10.2: Writing

When writing data an 8 bit command is written by the host processor followed by an 8 bit data field.

When a Motorola, National Semiconductor, or other host microprocessor is used, an SPI or MICROWIRE bit rate of 62,500 or less must be used in order to prevent overrunning the EIOC. Higher bit rates (maximum 2.1MHz) can be used if a delay is inserted between the write command and the write data byte when writing data to the EIOC. The additional delay may have to be as much as the READ DELAY if the bit rate is very high.

Here is an example Neuron C function for writing data to the EIOC:

```
IO_8 neurowire master select (IO_7) kbaud(20) IO_eioc;
IO_7 output bit IO_eioc_cs;

unsigned char eioc_data;

//
// This function writes a "write" command to the EIOC followed by a data
// byte.  The first function parameter is the command, the second is the
// data byte.
```

```
//
void wr_eioc(unsigned char wr_cmd,unsigned char wr_data)
{
  eioc_data=wr_cmd;
  io_out(IO_eioc,&eioc_data,8);
  eioc_data=wr_data;
  io_out(IO_eioc,&eioc_data,8);
}
```

3.10.3: Reading, READ DELAY

When reading data the host writes an 8 bit command, waits for a period of time (the READ DELAY), and then reads an 8 bit data field. The length of the READ DELAY is defined as follows:

1) If you are not using the IBM PC/AT keyboard interface, then READ DELAY must be at least 200µs.

2) If you are using the IBM PC/AT keyboard interface, but you are not using the asynchronous serial interface, then READ DELAY must be at least 250µs.

3) If you are using the IBM PC/AT keyboard interface, and you are using the asynchronous serial interface, then READ DELAY must be at least 300µs.

Here is an example Neuron C function for reading data from the EIOC:

```
//
// comment out the two READ_DELAY definitions that don't apply to you:
//
//#define READ_DELAY 1 /* 200us for no PC/AT kbd         */
//#define READ_DELAY 3 /* 250us for PC/AT kbd, no serial */
#define READ_DELAY 5 /* 300us for PC/AT kbd and serial */

IO_8 neurowire master select (IO_7) kbaud(20) IO_eioc;
IO_7 output bit IO_eioc_cs;

unsigned char eioc_data;

//
// This function writes a "read" command to the EIOC and then reads one byte
// from the EIOC which is returned by the function.
//
unsigned char rd_eioc(unsigned char rd_cmd)
{
  eioc_data=rd_cmd;
  io_out(IO_eioc,&eioc_data,8);
  if(rd_cmd==0x51)
    delay(33); /* delay 1ms for MMIO read data */
  else
    delay(READ_DELAY);  /* delay as needed for all other read commands */
  eioc_data=0;
  io_in(IO_eioc,&eioc_data,8);
  return(eioc_data);
}
```

3.11: EIOC Commands

This section contains a command summary table followed by a description of each command. For read commands the constant READ DELAY is defined in section 3.10.3: "Reading, READ DELAY"

EIOC Command Table

Command	R/W	Object
0xA5	write	synchronization
0x40	read	EIOC status register
0x41	read	EIOC main interrupt mask
0x61	write	EIOC main interrupt mask
0x42	read	EIOC main interrupt source
0x43	read	EIOC configuration register
0x63	write	EIOC configuration register
0x44	read	card reader data
0x45	read	EIOC status signals
0x46	read	keyboard data
0x67	write	keyboard reset
0x48	read	keyboard typematic/delay
0x68	write	keyboard typematic/delay
0x49	read	matrix keypad configuration
0x69	write	matrix keypad configuration
0x4A	read	real time clock year
0x6A	write	real time clock year
0x4B	read	real time clock month
0x6B	write	real time clock month
0x4C	read	real time clock date
0x6C	write	real time clock date
0x4D	read	real time clock day of week
0x6D	write	real time clock day of week
0x4E	read	real time clock hour
0x6E	write	real time clock hour
0x4F	read	real time clock minute
0x6F	write	real time clock minute
0x50	read	real time clock second
0x70	write	real time clock second
0x51	read	MMIO data
0x71	write	MMIO data
0x52	read	MMIO address
0x72	write	MMIO address
0x53	read	serial data
0x73	write	serial data
0x54	read	serial configuration register
0x74	write	serial configuration register
0x55	read	serial status register
0x56	read	string termination character
0x76	write	string termination character

cont...